

# Assignment 1

## Overview

This is an individual assignment that requires you to design, develop and test a small text-based program

## Timelines and Expectations

Percentage Value of Task: 20%

Due: 5 pm Friday Week 7

Minimum time expectation: 20 hours

## Learning Outcomes Assessed

The following course learning outcomes are assessed by completing this assessment:

### Knowledge:

- K1. Identify and use the correct syntax of a common programming language.
- K2. Recall and use typical programming constructs to design and implement simple software solutions.
- K4. Explain the importance of programming style concepts (documentation, mnemonic names, indentation).

### Skills:

- S1. Utilise pseudocode and/or algorithms as a major program design technique.
- S2. Write and implement a solution algorithm using basic programming constructs.
- S4. Describe program functionality based on analysis of given program code.

### Application of knowledge and skills:

- A1. Develop self-reliance and judgement in adapting algorithms to diverse contexts.
- A2. Design and write program solutions to identified problems using accepted design constructs.

## Assessment Details

The Federation University Australia Robot Wars are about to commence! The competitors are busy training and now it's time to prepare to sell spectator tickets.

Your task is to design, develop and test a small application for purchasing and recording ticket sales to this event

The assignment is broken up into three main components:

- 1) Design and model two classes: **Ticket** and **Checkout**
- 2) Create an **activity chart** which describes the behavior of the ticket sales system
- 3) Create a **computer program** that allows a user to interactively purchase a number of tickets, then provides an opportunity to enter some (fake) credit card information to pay for the tickets, and finally, prints out a receipt for the user (to the screen, not on paper). Finally, the program should backup all the transactions into a text file

Your submission should consist of a Word document containing the first two parts of the assignment, and three Python scripts that implement the computer program (**ticket.py**, **checkout.py** and **main.py**).

The **main.py** script runs the main logic of the program and will use instances of Ticket and Checkout classes to simulate purchasing a spectator ticket.

### Part 1: Design and Model Two Classes

This stage requires you to prepare documentation that describes the function of the program and how it is to be tested. There is no coding or code testing involved in this stage.

Requirements:

- 1) Read all of this assignment sheet first!!!
- 2) Write an algorithm that describes how the program will operate.
  - a. All program requirements must be included, even if you do not end up including all these requirements in your program code.
  - b. The algorithm must be structured logically so that the program would function correctly.
- 3) Create class diagrams for the Ticket and Checkout classes

Add everything to your Word document.

### Part 2: Activity Flowchart

Using either the online website <https://draw.io> (preferred), or the applications Visio or PowerPoint – create an activity diagram of how the program should operate to successfully purchase one or more tickets, accept payment, print a receipt for the user and backup the transaction to a text file.

Make sure to use the correct symbols in your diagram for starting, processes, decisions/branches, and ending the process.

Remember to do some (fake) checks when the user enters their credit card details e.g. that enough numbers have been entered.

Once you have completed your activity flowchart, add it to your Word document.

### **Part 3: Computer Program**

You are free to design and implement the software however you see fit. Here are some requirements that must be incorporated into your program

1. You must display a welcome message when the program starts. At a minimum, this message should contain the name of your program, the name of the program developer and your student ID.
2. The welcome message should also have a row of asterisks at the top and the bottom, just long enough to extend over the text. *Hint: Use a For loop for this.*
3. When the user goes to purchase tickets, they should be able to purchase a child, adult, senior or concession ticket(s). Each ticket category should cost a different amount; it's up to you to choose how much each ticket costs. They should be able to purchase as many tickets (across all categories) as they like
4. When the user goes to finalise their order, the total cost should display on screen. Your program should then ask the user to enter their (fake) credit card details, check the credit card details and, if 'valid', display a final receipt
5. Finally, the program should backup the transaction into a text file

## **Submission**

Your Word document and program code be zipped into a single file (a .zip file) and loaded into the Assignment Box provided in Moodle by the due date and time. The naming conventions for the zip file are:

**ITECH1400\_Assignment\_1\_<YOUR-NAME>\_<YOUR-STUDENT-ID>.zip**

Obviously replace <YOUR-NAME> and <YOUR-STUDENT-ID> with your own personal details!

Assignments will be marked on the basis of fulfilment of the requirements and the quality of the work. In addition to the marking criteria, marks may be deducted for failure to comply with the assignment requirements, including (but not limited to):

- Incomplete implementation(s)
- Incomplete submissions (e.g. missing files)
- Poor spelling and grammar

The mark distribution for this assignment is explained on the next page—please look at it carefully and compare your submission to the marking guide.

## Marking Criteria/Rubric

Task	Available Marks	Student Mark
<b>Stage 1: Design and Model Two Classes</b> Development of an algorithm describing how the program should function <ul style="list-style-type: none"> <li>All requirements from the assignment sheet are included</li> <li>Logical structure</li> </ul> Development of class diagrams <ul style="list-style-type: none"> <li>All requirements from the assignment sheet are included</li> <li>Class diagrams are correctly formatted</li> </ul>	2 2 1 1	
<b>Stage 2: Activity Flowchart</b> Creation of an activity flowchart which clearly indicates how the program should operate, using the correct symbols for elements such as start/end points, processes and decision/branches	2	
<b>Stage 3: Computer Program</b> <ul style="list-style-type: none"> <li>Welcome message (including looping structure)</li> <li>Creates four Ticket instances (child, adult, senior, concession) that may be purchased (student to nominate own pricing schedule)</li> <li>Adds Ticket to the Checkout list of tickets being purchased</li> <li>Allows the checkout of multiple tickets</li> <li>Accepts simulated 'credit card' details (including instances when the cards will be refused e.g not enough numbers entered)</li> <li>Prints a final receipt of the tickets purchased, along with the total cost</li> <li>Stores the transaction into a file</li> </ul>	1 1 2 2 2 2 2	
Total	20	

## Feedback

Assignments will be marked within 2 weeks of submission. Marks will be loaded in fdlGrades, and a completed marking sheet will be available via Moodle.

## Plagiarism:

Plagiarism is the presentation of the expressed thought or work of another person as though it is one's own without properly acknowledging that person. You must not allow other students to copy your work and must take care to safeguard against this happening. More information about the plagiarism policy and procedure for the university can be found at <http://federation.edu.au/students/learning-and-study/online-help-with/plagiarism>.